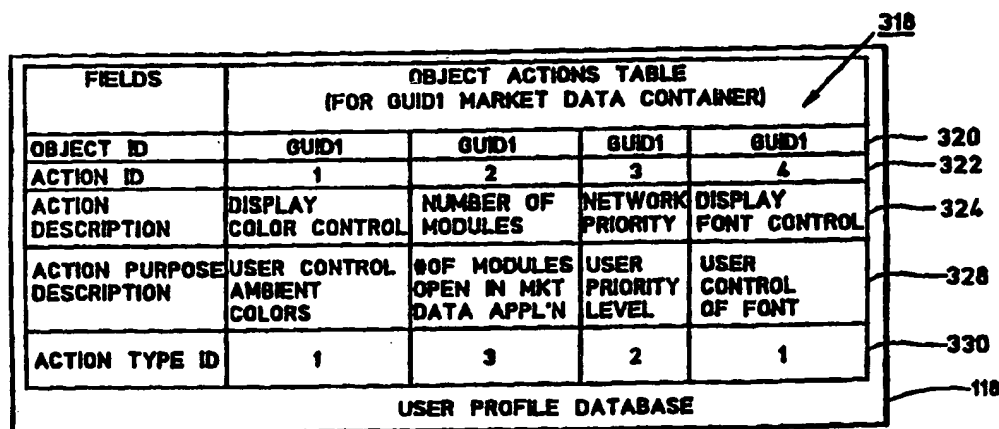




## INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

<p>(51) International Patent Classification <sup>6</sup> : <b>G06F 1/00, 3/033, 17/60</b></p>	<b>A1</b>	<p>(11) International Publication Number: <b>WO 99/40502</b></p> <p>(43) International Publication Date: 12 August 1999 (12.08.99)</p>						
<div style="display: flex; justify-content: space-between;"> <div style="width: 55%;"> <p>(21) International Application Number: <b>PCI/US99/02709</b></p> <p>(22) International Filing Date: 9 February 1999 (09.02.99)</p> <p>(30) Priority Data:</p> <table style="width: 100%; border: none;"> <tr> <td style="width: 30%;">60/074,142</td> <td style="width: 30%;">9 February 1998 (09.02.98)</td> <td style="width: 40%;">US</td> </tr> <tr> <td>09/246,351</td> <td>8 February 1999 (08.02.99)</td> <td>US</td> </tr> </table> <p>(63) Related by Continuation (CON) or Continuation-in-Part (CIP) to Earlier Application</p> <p style="margin-left: 40px;">US 60/074,142 (CON)</p> <p style="margin-left: 40px;">Filed on 9 February 1998 (09.02.98)</p> <p>(71) Applicant (for all designated States except US): <b>REUTERS, LTD. [GB/GB]; 85 Fleet Street, Greater London, London EC4P 4AJ (GB).</b></p> <p>(72) Inventors; and</p> <p>(75) Inventors/Applicants (for US only): <b>CHANDRA, Vipin [IN/US]; Unit 8, 2329 Kansas Avenue, Santa Monica, CA 90404 (US). KLIMCZAK, Jarek [US/US]; 5616 Azure Way, Long Beach, CA 90803 (US).</b></p> <p>(74) Agent: <b>MAXHAM, Lawrence, A.; Baker &amp; Maxham, Suite 3100, 750 B Street, San Diego, CA 92101 (US).</b></p> </div> <div style="width: 40%;"> <p>(81) Designated States: <b>AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).</b></p> <p><b>Published</b></p> <p><i>With international search report.</i></p> <p><i>Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i></p> </div> </div>			60/074,142	9 February 1998 (09.02.98)	US	09/246,351	8 February 1999 (08.02.99)	US
60/074,142	9 February 1998 (09.02.98)	US						
09/246,351	8 February 1999 (08.02.99)	US						

(54) Title: METHOD OF CONTROLLING SOFTWARE APPLICATIONS SPECIFIC TO A GROUP OF USERS



## (57) Abstract

Computer software application which defines "action items" in order to allow subscriber control of user access to various functionality or features of the application on a user-by-user basis. "Action items" may correspond to many various types of functionality relating to the user interface, such as data display, data output and data transfer, rather than to database access or database modification privileges. In this way a subscriber can give its users access to more or fewer user interface features, depending upon individual user needs and abilities, on an individualized basis.

**FOR THE PURPOSES OF INFORMATION ONLY**

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon		Republic of Korea	PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

## METHOD OF CONTROLLING SOFTWARE APPLICATIONS SPECIFIC TO A GROUP OF USERS

### TECHNICAL FIELD

5 This invention relates generally to computer software applications providing for customized access to software applications, and more particularly to subscriber control of the customization on a user-by-user basis. Note that a subscriber is an institutional software customer, such as a company, and a user is an individual (person or piece of hardware) which uses the application supplied by the subscriber. For example, application users might be company employees who use an application in the course of their employment with the subscriber.

### BACKGROUND ART

10 In some conventional software applications, such as word processing programs, web browsers and market data applications (for receiving current financial information), users can customize aspects of the user interface. For example, a user may be able to customize application display colors, or print format, or data transfer rate of data transfers, or the content of pull down menus and so on.

15 One advantage of this kind of application is that a sophisticated and disciplined users can optimize their interfaces according to their individualized wants and needs. On the other hand, a disadvantage of these conventional applications is that an unsophisticated or undisciplined user might customize her user interface in a counter-productive way. This is not good for the user or for her associated subscriber.

### 20 DISCLOSURE OF INVENTION

It is an object of at least some embodiments of the present invention to provide for subscriber control of the user interface for users associated with the subscriber. It is a feature of at least some embodiments of the present invention that a software application defines action items relating to aspects of each user's access, which action items are  
25 configured by the subscriber on a user-by-user basis. It is an advantage of at least some

embodiments of the present invention that users are allowed access to aspects of the user interface on an individualized basis according to subscriber's configuration decisions.

According to the present invention, a method of using a software application generally includes the steps of providing an application having "action items" to a subscriber, and having the subscriber configure the "action items" with "action item values" on a user-by-user basis (or on the basis of groupings of users).

As the term is used herein, an "action item" is any aspect of the functionality of the software application, except that "action items" do not include access to data or databases, access to portions of data or databases, ability to modify data or databases, or ability to control action item configuration or data or database modification permission. To illustrate, providing for control of application display colors is an example of an "action item," while access to some specific database which can be provided by the application cannot be considered as an "action item." Other examples of "action items" will be given throughout this application to help convey the concept of "action items," but these examples are not intended to limit the definition of "action items" provided in this paragraph.

A physical medium can be coded with a computer software application according to the present invention. Some exemplary physical mediums are floppy disks, hard magnetic disks, optical disks, magnetic tapes, read only memories, random access memories and so on. The coded software application includes application code and configuration code. The application code corresponds to computer instructions for executing the software application, with the application code defining at least one action item. The configuration code corresponds to computer instructions for configuring the computer software application with action item values the users, whereby each user's access with respect to the action item is determined by the action item value. This way, a subscriber can configure the application for each user by defining the action item values.

It is noted that while the present invention relates to subscriber (institutional software customer) control, and more particularly to subscriber (institutional software customer) control effected by configuring software, this configuration will generally be carried out by an individual or individuals who have been delegated to decide on the subscriber's configuration strategy and to actually input the desired configuration information into the appropriate computer(s). Therefore, as used herein, the term

"subscriber" includes these delegated individuals. In some embodiments of the present invention, the delegated individuals may include third party individuals, user individuals, or individuals affiliated with the software manufacturer.

### **BRIEF DESCRIPTION OF THE DRAWING**

5           The objects, advantages and features of the present invention will become more readily apparent from the following detailed description, taken together with the accompanying drawing, in which:

Fig. 1 is a block diagram of an embodiment of a computer system which stores and executes an embodiment of software according to the present invention;

10           Fig. 2 is a levels set table used by the software of Fig. 1;

Fig. 3 is an objects set table used by the software of Fig. 1;

Fig. 4 is a level-objects table used by the software of Fig. 1;

Fig. 5 is an object actions table used by the software of Fig. 1;

Fig. 6 is an action value type set table used by the software of Fig. 1;

15           Fig. 7 is an actions type table used by the software of Fig. 1;

Fig. 8 is a main profile table used by the software of Fig. 1;

Fig. 9 is a block diagram of a portion of the software of Fig. 1; and

Fig. 10 is a flowchart representing operations of the software of Fig. 1.

### **BEST MODE FOR CARRYING OUT THE INVENTION**

20           The present invention relates generally to any kind of software which can be purchased by a first party subscriber (for example, person, company or other entity) and then used by a plurality of second party users (for example, employees of a company). According to the present invention, the subscriber can set the software up differently for each of the users on an individualized basis (or on the basis of groupings of users). More  
25           particularly, the first party can set action item values to individually control action items for each of the users. The action items may relate to many different parameters of the software operation, such as extent of user control of the application display, extent of user control of the data output, or extent of user control of data transfer.

Below, the present invention will be explained in terms of an exemplary embodiment of a market data software package for providing current and historical financial information to brokers. In this example, the subscriber will generally be a brokerage house which can customize the market data software for each of its broker-users. Each broker-user will have a set of action item values to control action items. These action items will determine various aspects of how the software will be executed for each broker-user. In this way, the subscriber can set up the software so that each broker-user can use the software in an optimal fashion according to the abilities and needs of each individual broker-user.

Fig. 1 shows an embodiment of a computer system 100 which can be used to run the exemplary market data software according to the present invention. For the sake of simplicity, exemplary computer system 100 is a stand-alone personal computer (PC) system. However, in some preferred embodiments, the market data software may be operated on a computer system having one or more server computers and one or more user workstations. This kind of server/workstation architecture is typical in many brokerage houses. The server computers store data or programs which is be accessed by users (such as brokers) through individual workstations, such as PC's.

Computer system 100 includes user PC 102, mouse 112, keyboard 114, display 116 and printer 117. Mouse 112 is a conventional mouse input device which is connected to provide user input to user PC 102. Keyboard 114 is a conventional keyboard input device which is connected to provide user input to user PC 102. Display 116 is a conventional monitor which provides a screen display, such as a cathode ray tube display or liquid crystal display, which includes an application display generated by market data software 118 and output by user PC 102. Printer 117 is a conventional printer which provides a hard copy, such as a printed page of data output from user PC 102.

User PC 102 includes processing module 104, storage device 106, network card 107 (with networking software) and random access memory (RAM) 108. Processing module 104 provides the data processing functions of the user PC 102. Processing module 104 would include a conventional central processing unit and other conventional data processing circuitry. Storage device 106 provides permanent data storage. Storage device

106 may be, for example, a hard magnetic disk. RAM 108 provides temporary storage of data. RAM 108 may be a conventional RAM memory.

5 Network card 107 can communicate with network 200. Priority given to communications to and from network card 107 are controlled according to a variable priority level. For example if the priority value of network card 107 is set to 1 (low), then the communications may be handled as relatively low priority communications. Priority values 2 and 3 correspond respectively to medium and high priority level communications. Network 200 may be a public network of computers, such as the Internet, a private network of computers, such as an office intranet, or it may be a combination of these two kinds of  
10 networks.

As shown in Fig. 1, computer files making up the market data software package 118 are stored in storage device 106. Alternatively, these files could be stored wholly or in part on a removable storage device, such as a CD-ROM, or on a separate server computer. When market data software 118 is operated by a user, data related to program execution is  
15 also stored in RAM 108. Below, a portion of market data software 118 called user profile database 120 will be discussed in some detail.

User profile database 120 is created and maintained by the user profile system 122, which is a security sub system in market data software 118. In this example, user profile database 118 contains data for all users who might potentially use the user PC 102. In  
20 embodiments where there is a server computer, the server computer may store and maintain a user profile database for every possible user.

One exemplary user profile system is structured as an extension to an existing policy system that was used with Quotron (now called Reuters Plus) market data software package release version 5.01.07 and earlier versions. The user profile system 122 is used to  
25 configure user profile database 120 to control user access parameters by the subscriber on a per user basis. In this example, each user is assigned a security level (expert, medium or novice) which corresponds to three different ways that action items are configured. The security level represents the extent of access privileges for groups of users as controlled by the action item configuration.

30 Alternatively, all action item values could be uniquely configured for each and every user. While this alternative scheme allows greater flexibility, grouping users into a few

different security levels makes it much less time consuming for the subscriber to configure user profile system 122 when there are a great multitude of users. This is because each user merely needs to be assigned a single security level for predetermined access, rather than being configured for custom settings with respect to every action item.

5           The market data software 118 includes many "objects." "Objects" is herein used as a generic word to refer any module (Executables, Agents or AAE Service OCXs) that is participating in the market data software 118. Each object is identified with a unique ID called the Object ID. Action items (defined above) are properties or attributes of an object that can be configured using user profile system 122. Action items are the items of the  
10           object which are imposed with some sort of access control using user profile system 122.

          Implementation of market data software 118 proceeds in three stages. First, at the design stage, action items are identified and classified and user profile system 122 is written accordingly. Second, at the subscriber configuration stage, action item values are set by the subscriber for each of the users so that the market data software 118 will operate in a  
15           customized way for each security level. Third, at the run time stage, the user operates market data software 118, with the action item parameters being controlled according to the subscriber-set action item values for that particular user. Each of these stages is discussed in more detail below.

#### 1.     Design Stage

20           Generally, the market data software will not be designed on user PC 102, but rather will be designed by a software designer on some other machine, from where the software is copied to storage device 106. The design time stage involves creation of user profile database 120. Design activity involves each of the modules (objects) in the market data software 118, as each module has to generate the parameters which are to be subscriber-  
25           controlled "action items" that will be participating in the user profile system.

          Generation of the action items information for a module will now be described. Action information is stored in the serialized files with extension ".PRO." Each module has to provide the following pieces of information in order to generate the .PRO file:  
Object ID: This is an identification code (GUID) associated with the module,  
30           Object Description: Name of the object, and



Array of Action Items: This contains following information:

Action ID: Unique integer value to identify an action item for this object,

Action Desc: Description of the action item. This could be name of the menu item or a particular attribute of the object, and

5 Action Item Type: This field tells the type of the action item.

Action item types are an important aspect of some embodiments of the present invention, because the use of different action items types allow different kinds of user access parameters to be efficiently handled through a single user profile system 122. Each defined action item will be classified under one of the predetermined action item types. In  
10 this exemplary embodiment, three types of actions are identified:

(1) Dual State Action Type: This action type is used to represent any action item that has two states -- TRUE/ENABLE and FALSE/DISABLE.

(2) Limit Action Type: This is used to represent those properties (actions) of the object that can be best described with an upper or lower limit.

15 (3) Range Action Type: This is used to represent those properties (actions) of the object that can best be described by having a lower and an upper limit.

Apart from these types of actions more generic action types can be defined.

One example of an action item is the "menu" action item, which is a Dual State Action Type. Another example is "object count" action item, which is a Limit Action Type.

20 Another piece of information used in creating the .PRO file is the Action Purpose Description. The Action Purpose Description gives the detailed description about the action item, which is displayed in the profile editor when the subscriber selects the action item for configuration. The Action Purpose Description should give a clear description of the action item and its purpose.

25 Another piece of information used in creating the .PRO file is the Default Value. The Default Values are stored under the default security level. When an object requests the profile information for an action item, but the action item is missing, then these default values are returned as the values for the action item. Values should be set with an Action Default Value and an Action Default Extra Value.

30 With respect to the Action Default Value, if an action item is a menu item or a Dual State Action Type item, then set it as either TRUE or FALSE. If the action item is a Range

Action Type item, then it is set as the default minimum range. For Limit Action Type items, the Action Default Value is ignored.

5 With respect to the Action Default Extra Value, for Dual State Action Type items, the value set to this file is ignored. For Limit Action Types items, the Action Default Extra Value sets the default limit for the action item. For Range Action Type action items, this filed should be set to a value which will be treated as the default maximum value. Preferably, the maximum default value (Action Default Extra Value for Range item) is set to be less than the minimum default value (Action Default Value for Range item).

10 The foregoing pieces of information are maintained internally by each module and used to generate the serialized action information file with a .PRO extension. One example of a possible action item defined by various modules is user control of the application display colors. This may defined as a Dual State Action Type which can be configured to ENABLE (user permitted to control display colors) or DISABLE (user not permitted to control display colors).

## 15 2. Subscriber Configuration Stage

After the subscriber installs the market data software 118 on user PC 102, the subscriber uses a tool called Profile Editor to configure and update the user profile database 120, so that each broker-user will be able to optimally operate the market data software on an individualized basis.

20 The Profile Editor provides a graphical user interface to set action item values for each user for each of the action items. Using this tool, different security levels can be created. Interface is provided to import the action information generated by each of the modules. Using this imported action information, user profile database 120 is updated for the object of interest. The detailed control flow for user profile database 120 updates will be explained below.

25 Returning to the exemplary action item relating to user control of application display colors, the subscriber sets an action item value for this action item as ENABLE or DISABLE for each user, depending upon whether the subscriber perceives that the particular user is talented and disciplined enough to utilize control of application display colors in a constructive manner.

30

It is noted that user control of application display colors (as well as other action items) may be restricted for somewhat subtle reasons. For example, some financial information may be necessarily displayed in a predetermined color which cannot be changed. In this case, if a user set the ambient background color of the application display to be the same as the predetermined color, then the information in the predetermined color would not be visible in the application display. Clearly, this would be a problem because the user would miss information, and might not even be aware that the hidden information was not being visibly displayed.

### 3. Run Time Stage

At this stage, a user operates the market data software 118, which will be controlled in accordance with the action item values set by the subscriber for the security level of the particular user. When a module is started, user profile information is requested from the user profile database 120. User profile database 120 is consulted for the requested module under the security level assigned to the user.

Preferably, the user who is using the market data software 118 will be identified by entering a user identification code and a password. However, there are other ways to make sure that the correct user profile information is consulted for the user. For example, each user may have a dedicated workstation so that the workstation will have only user profile information for some predetermined user. This may make it unnecessary to use user identification codes or the like. As another alternative, user identity could be determined by conventional electronic fingerprinting, photographing, voice recognition or the like.

Now that the three stages of action item implementation have been described, various tables used to information related to action items and action item values will be described in more detail. For the sake of simplicity, the tables of this example will be have information for only a few users, objects, action items, and so on. Actual tables may be much larger to accommodate all of the necessary action item related information. User profile database 120 is used to maintain the action items values for the users. User profile database 120 is created using Microsoft Access 97. The tables discussed directly below are defined to maintain the user profile information.

Levels set table 300 is shown in Fig. 2. Levels set table 300 is used to maintain all currently available security levels. It has two fields, Level ID field 302 and Level Description field 304. Level ID 302 is an integer value which uniquely identifies the security level. Level Description is a user friendly name assigned to the security level. Initially, when an empty database is created, an entry called "Default Level" is added to the database.

As shown in Fig. 3, objects set table 306 is used to maintain a list of the objects that are participating in user profile system 122. In other words, this table keeps track of all objects which have associated action item parameters. Objects set table 306 has two fields, Object ID field 308 and Object Description field 310. Object ID 308 is a Globally Unique Identifier (GUID) used to represent each object. Object description 310 is a user friendly name assigned to each object. When a database is created, a default entry is added to this table with the description "Default Object."

As shown in Fig. 4, level-objects table 312 is used to maintain lists of objects under each security level. Level-objects table 312 has two fields, Level ID field 314 and Object ID field 316. Object ID 316 is of the GUID for each object. Level ID 314 is an associated integer value uniquely identifying the security level for each object. There can be more than one object under each security level. When a database is created an entry is created in this table for the "Default Object" under the "Default Level."

As shown in Fig. 5, object actions table 318 is used to maintain list of available action items under each object that participates in the user profile system 122. This table has five fields: (1) Object ID 320, (2) Action ID 322, (3) Action Description 324, (4) Action Purpose Description 328 and (5) Action Type ID 330. Object Id 320 is the GUID used to represent each object. Action Id 322 is an integer uniquely identifying each action item. Action Description 324 is a description of each action item and is object specific. Action Purpose Description 328 is a clear description of the purpose served by each action item. Action Type Id 330 indicates the action type of action item and is indexed to an entry in the Actions Type Table discussed below.

As shown in Fig. 6, actions value typeset table 332 is a fixed table with three entries. As explained above, in this example, three types of actions are supported based on the values they take. They are:

(1) Dual State Actions -- this type supports Boolean values, such as ENABLE or DISABLE,

(2) Limits Type Actions -- this type supports an upper limit value, and

(3) Range Action Type -- this type supports lower and upper limit values.

5 The actions value type table has two fields, Action Value Type ID field 334 and Action Value Type Description 336. Action Value Type Id 334 identifies the action value type. Action Value Type Description 336 describes the value type, such as "Dual State," "Limit," or "Range".

10 As shown in Fig. 7, actions type table 338 is a read only table with a fixed number of entries. The actions type table 338 is used to support more user friendly action types. This table is one level higher than actions value type table 332. The idea behind introducing this table above the action value type table 332 is to categorize actions based on the purpose they serve rather than on the values they support. Entries of the action type table 338 are associated with entries in the actions value type table depending on the type of values they support.

15 Actions type table 338 has three fields: (1) Action Type ID 340, (2) Action Type Description 342, and (3) Action Value Type ID 344. Action Type ID 340 identifies an action type. Action Type Description 342 describes the action type in a user-friendly manner. Action Value Type Id 344 is an index to an entry in the action value type table 332.

20 In this example, five entries (action types) are supported as given below. As menu item supports two possible values (ENABLE/DISABLE) in the user profile system context, it can be viewed as an action type that supports two states (that is, a Dual State Action). So the Action Value Type ID field 340 is set to 1 for menu item.

25 If new action types are identified which have a general meaning for more than one module, then a further entry can be added to the actions type table 338. For example, an action type called "Max Instance" could be added as another action type if this action type is used in more than one module. On the other hand, if this type of action makes sense only in the context of one module (Object), then it can be described with the action type "Generic Limit Action" (Action Type ID = 4). As described above, a module has to set the action type for each of the actions it is intended to include in user profile system 122. The

value exported (through the .PRO file) as action type should correspond to one of the entries in this table.

As shown in Fig. 8, main profile table 346 is used to store action item values for a security level. Main profile table 346 maintains the list of action items that have been declared to be participating in the user profile system 122 by the subscriber. This is information used by the user profile system 122 when a user runs the market data software 118.

Main profile table 346 has five fields: (1) Level ID field 348, (2) Object ID field 350, (3) Action ID field 352, (4) Action Value field 354, and (5) Action Extra Value field 356. Each entry in main profile table 346 corresponds to an action item with values set under an object for the security level.

Level ID 348 corresponds to the security level to which the action items values are of interest. Object ID 350 identifies the module to which an action item belongs. Action ID 352 identifies the action item.

Action Value 354 is a first action item value set for an action item by the subscriber. The meaning of the value set to this field depends on the action item type of the action item. For Dual State Action Type action items, Action Value 354 will be a binary value (for example ENABLE or DISABLE). For Limit Action Type Action items, the Action Value 354 is ignored. For Range Action Type action items, the Action Value 354 is set to some minimum (or maximum) value used for controlling the action item parameter.

Action Extra Value 356 is additional information which may be required by an action item. The meaning of Action Extra Value 356 set to this field depends on the action item type of the action item. For Dual State Action type action items, the Action Extra Value 356 is not needed. For Limit Action Type Action items, the Action Extra Value 356 is value set as a limit of the action item. For Range Action Type action items, the Action Extra Value 356 is set to some maximum value used for controlling the action item parameter.

Although the embodiment discussed in detail herein includes two fields, Action Value 354 and Action Extra Value 356, for each action item, in many preferred embodiments of the present invention, only a single field is used to store the value of the action item. For example, an Action Item Value field could be set up to store Dual State

Action Type action item values as binary numbers, to store Limit Action Type action item values as variants representing an upper or lower limit, and to store Range Action Type action item values as variants each representing both an upper and a lower limit.

Now generation of an action information file (.PRO file) will be described. Action information about an object might change during the course of enhancements to the objects. New action items might need to be added to user profile database 120 or some of the actions might become obsolete. In order to support such changes to user profile database 120 and to safeguard the overall system after the changes are made, the concept of generation of action information by each object is utilized.

All the objects participating in the user profile database 120 should define a set of actions that can be configured in the user profile system 122. The objects should maintain full information about the action internally. A simple structure can be used to maintain the action information. An example of code for this kind of simple structure follows:

```
typedef struct
{
    int    nActionID;
    char   csActionDesc[MAX_ACTION_DESC_LEN];
    char   csPurposeDesc[MAX_ACTION_PURPOSE_DESC_LEN];
    int    nActionType;
    long   nInternalRefId; // This used for internal reference in the context of the
                           object.
} ProfileActionInfo;
```

As an example, entries appropriate to an application could be provided as shown in the following code:

```
ProfileActionInfo ProfileInfo[] =
{
    {1,  "File New", "A menu item to create a new Nav Document",
     ACTION_MENUITEM,
     ID_FILE_NEW},
    {2,  "File Open", "A menu item to open an existing Nav Document",
```

```

ACTION_MENUITEM, ID_FILE_OPEN},
{3,  "File Print", "Provision to print a Nav document",  ACTION_MENUITEM,
    ID_FILE_PRINT},
{4,  "File Exit", "Provision to exit Navigator through menu option",
5    ACTION_MENUITEM, ID_APP_EXIT},
{5,  "Edit", "Enable or disable the entire editing functionality", ACTION_MENUITEM,
    ID_POPUP},
{6,  "Insert", "Provision to insert AAE services or any Ole Objects & to create add new
    navigator pages",  ACTION_MENUITEM,  ID_POPUP},
10 {7,  "Move Item", "Ability to move any object in the navigator. If it is set to FALSE,
    objects cannot be moved inside the Navigator.", ACTION_DUALSTATE,
    ID_PROP_MOVE_ITEM},
{8,  "Resize Item", "Ability to resize the objects in the navigator. If it is set to FALSE,
    objects cannot be resized inside the Navigator.", ACTION_DUALSTATE,
15 ID_PROP_RESIZE_ITEM},
{9,  "Objects per Page", "This sets a limit as to how many objects can be inserted in to
    a Navigator page", ACTION_LIMIT,  ID_PROP_OBJECTS_PER_PAGE},
{10, "Max. Pages", "This sets a limit as what is the maximum number of pages that can
    be added to a navigator document", ACTION_LIMIT,  ID_PROP_MAX_PAGES},
20 {11, "Max Objects", "This is the total number of objects that are allowed to present in
    the navigator page". ACTION_OBJECT_COUNT, ID_PROP_MAX_OBJECTS},
{12, "Rename Pages", "Ability to rename a page item", ACTION_MENUITEM,
    ID_RENAME_PAGE}
};

```

25 Using this information the objects generate a .PRO file which is a serialized file having following structure (which includes an array of CProfileInfo objects where each object holds the information about each object):

```

class CProInfoArray : public CobArray
{

```



```

    virtual void Serialize(CArchive& ar) { Serialize ObjectID & Description & set of
    Action info. objects;}
    CString m_csObjecID;
    CString m_csObjectDesc;
5    };
class CProfileInfo : public CObject
{
    int    m_nActionID;
    CString m_csActionDesc;
10    CString m_csPurposeDesc;
    int    m_nActionType;
void Serialize(CArchive& ar) { Serialize Action ID, Action Description & Action type};
};

```

15 The generation of .PRO files for market data software OCX's can be done with a small application that calls a specific exported function in the OCX that generates the object file. OCX name can be passed as argument to this application. For generating the .PRO files for executables, like navigator, provision for a command line option should be supported. When the application is executed with a particular command line option it should generate the action information file.

20 The profile editor tool provides an interface for the subscriber to select a particular .PRO file and import the action information into user profile database 120. If an object corresponding to the .PRO file is already present in user profile database 120, then the subscriber is prompted with a message indicating that the previous action information will be replaced with the new action information and all the references to the old action
 25 information will be deleted. If subscriber agrees with that, then a new set of actions are defined under the selected object in the user profile database 120, replacing the old action information.

If object is new to the database, then new set of actions are defined under that object. In some embodiments according to the present invention, provision can be made
 30 to selectively import a single action item or to overwrite a single action item with a new

one. The criteria used to check for an existing action item is the check for duplicate Action ID's (GUID). Each object in its internal profile information structure defines the Action ID's.

5 Now control flow in the run time components stage will be discussed with reference to Fig. 9. As shown in Fig. 9, market data software 118 includes user profile database 120, three objects 400, 402, 404, QMKTDATA DLL 406, entitlement agent 408 and comm manager 410.

10 When the objects 400, 402, 404 are created, they request entitlement agent 408 for user profile information. In doing so, the objects 400, 402, 404 have to provide the user's security level (that is the group of users, expert, medium or novice to which the user belongs). Objects 400, 402, 406 must further provide Object ID and set of action items, whose values are pertinent to the object. Once the profile information is received from entitlement agent 408, each object 400, 402, 404 will configure itself according to the user profile information received.

15 Entitlement agent 408 is the heart of the user profile system 122. When a request for user profile information is received from an object, entitlement agent 408 executes a sequence of steps before sending the user profile information back to the requesting object.

20 These steps are explained below with reference to the flow chart of Fig. 10. At step S1, the request for user profile information and the security level are received. At step S2, entitlement agent 408 checks whether the specified security level is present in the user profile database 120. As explained above, user profile database 120 always contains information about a security level called "Default Level." The idea behind introducing a default level is to ensure that all objects gets some profile information upon their request to entitlement agent 408.

25 At step S3, entitlement agent 408 will look for user profile information under the default security level only if the security level assigned to the user is not available in the database.

30 At step S4 entitlement agent 408 determines whether the object is present under the requested level. Under each level at least one object called "Default Object" will always be present. When user profile information for a particular object is not available for the

requested level, then at step S5 the default Profile information is returned as available under the "Default Object," and processing proceeds to step S6.

At step S7 it is determined whether an action item is present. Apart from default level and default objects, user profile database 120 also supports the concept of default action values. User profile database 120 maintains default values for each type of action under each object. When an object request for the user profile information for a set of action item, entitlement agent 408 looks for the requested action items at step S7.

If an action item is not declared to be participating in the user profile system for that object under a given security level, then at step S8 entitlement agent 408 checks the type of the action item and returns the default value set for that type under that object. On the other hand, if the action item is participating, then entitlement agent 408 reads the values set for this action item. At step S10, entitlement agent 408 returns the user profile information obtained at step S8 or step S9 to the requesting object.

In order to more fully illustrate the concept of an action item (defined above) some exemplary kinds of action items will now be discussed. However, these examples should not be used to limit the definition of the term "action item."

One category of action item relates to the extent to which the user can control the application display portion of display 116. For example, action items can be defined to specify whether individual users (or groups of users grouped under various security levels) can control display colors or fonts of displays generated by market data software 118. These action items might be, for example, Dual State Action Type action items determining whether or not users can adjust display colors or fonts.

Another category of action items relates to the process of data transfer when executing the market data software. For example, the priority value (explained above) of communications through network card 107 may be set for example as a Limit Action Type action item, where the limit represents the maximum priority which a user can assign to her data transfer communications.

As shown in Fig. 5, the network priority for the market data container (GUID1) is controlled as a Limit Action Type (2). As shown in Fig. 8, the subscriber has configured the Action Extra Value for this action item to 2 as the maximum limit. This means that the user can control the priority value, but cannot set the priority value to be any higher than

2. In this way, the subscriber can help ensure that only extremely important data transfers are executed at the highest priority level 3.

Another category of action items relates to the process of data output when executing market data software 118. For example, the extent to which users can control print format of hard copy of data output through printer 117 can be controlled by the subscriber through the use of one or more appropriate action items.

Another category of action items relates to the input devices, such as keyboard 114 and mouse 112. For example, the subscriber could configure appropriate action items to predefine the functions of various mouse 112 buttons or keyboard 114 keys.

Another category of action items relates to aspects of the user's interface with market data software 118. For example, the content of pull down menus can be customized for each user (or each security level) using action items.

Also, as shown in Figs. 5 and 8, the number of financial modules which the user can insert in the broker application environment of market data software 118 can be controlled as a Range Action Type action item where the user can define between 4 and 25 modules. The greater the number of modules that a user inserts, the greater the demands on computer system resources and on communication bandwidth. Therefore, the number of modules allowed each user can be an important area for subscriber restrictions on the extent user control, because the subscriber can most accurately judge the relative needs of various users and can most fairly distribute limited system resources.

Certain preferred embodiments have been described above. It is likely that there are modifications and improvements to these embodiments which are within the literal scope or are equivalents of the claims which follow.

**CLAIMS**

What is claimed is:

1. A method of using a software application comprising the following steps:  
providing a copy of the application with at least one action item to a subscriber  
which has a plurality of associated users; and  
5 configuring the value of the action item for each user by the subscriber.
2. The method of claim 1 further comprising the step of executing of the  
software application by the users so that the functionality of the software will be controlled  
for each user in accordance with the value of the action item for that user.
- 10 3. The method of claim 1 wherein the action item is control of an aspect of a  
display when executing the application.
4. The method according to claim 3 wherein the action item is control colors  
of the display.
- 15 5. The method according to claim 1 wherein the action item is control of an  
aspect of a process of data transfer when executing the application.
6. The method according to claim 5 wherein the action item is control of the  
rate of data transfer.
7. The method according to claim 1 wherein the action item is control of an  
aspect of data output from the application.
- 20 8. The method according to claim 7 wherein the action item is control of hard  
copy print format of substantive data output by the application.
9. The method according to claim 1 wherein the action item is control of an  
aspect of the user interface with the features of the application.

10. The method according to claim 9 wherein the action item is the content of pull-down command menus which the user can use to access or manipulate the substantive data.

5 11. The method according to claim 1 wherein the action item is a dual state type action.

12. The method according to claim 1 wherein the action item is a limit action type.

13. The method according to claim 1 wherein the action item is a range action type.

10 14. The method according to claim 1 wherein the action item is an extent of control action item which controls an extent of control which the user has with respect to an aspect of operation of the application, whereby the user maintains some control over the aspect, but is also restricted in her control of the aspect according the value of the action item configured by the subscriber.

15 15. A method for subscriber control of broker access to features of a market data application, the method comprising the steps of:

coding by a market data software manufacturer a market data application which includes a plurality of action items;

providing a copy of the market data application to a brokerage house; and

20 configuring the value of the action item for each broker by the brokerage house.

16. A physical medium having a computer software application coded thereon, the medium comprising:

a substrate structured so that software can be coded thereon; and

code coded onto the substrate, the code comprising:

application code corresponding to computer instructions for executing the software application, with the application code defining at least one action item, and

configuration code corresponding to computer instructions for configuring the computer software application with action item values respectively for a plurality of users, whereby each user's access with respect to the action item is determined by the action item value.

17. The medium of claim 16 wherein the action item is control of an aspect of a display when executing the application.

18. The medium according to claim 16 wherein the action item is control of an aspect of a process of data transfer when executing the application.

19. The medium according to claim 16 wherein the action item is control of an aspect of data output from the application.

20. The medium according to claim 16 wherein the action item is control of an aspect of the user interface with the features of the application.

1/6

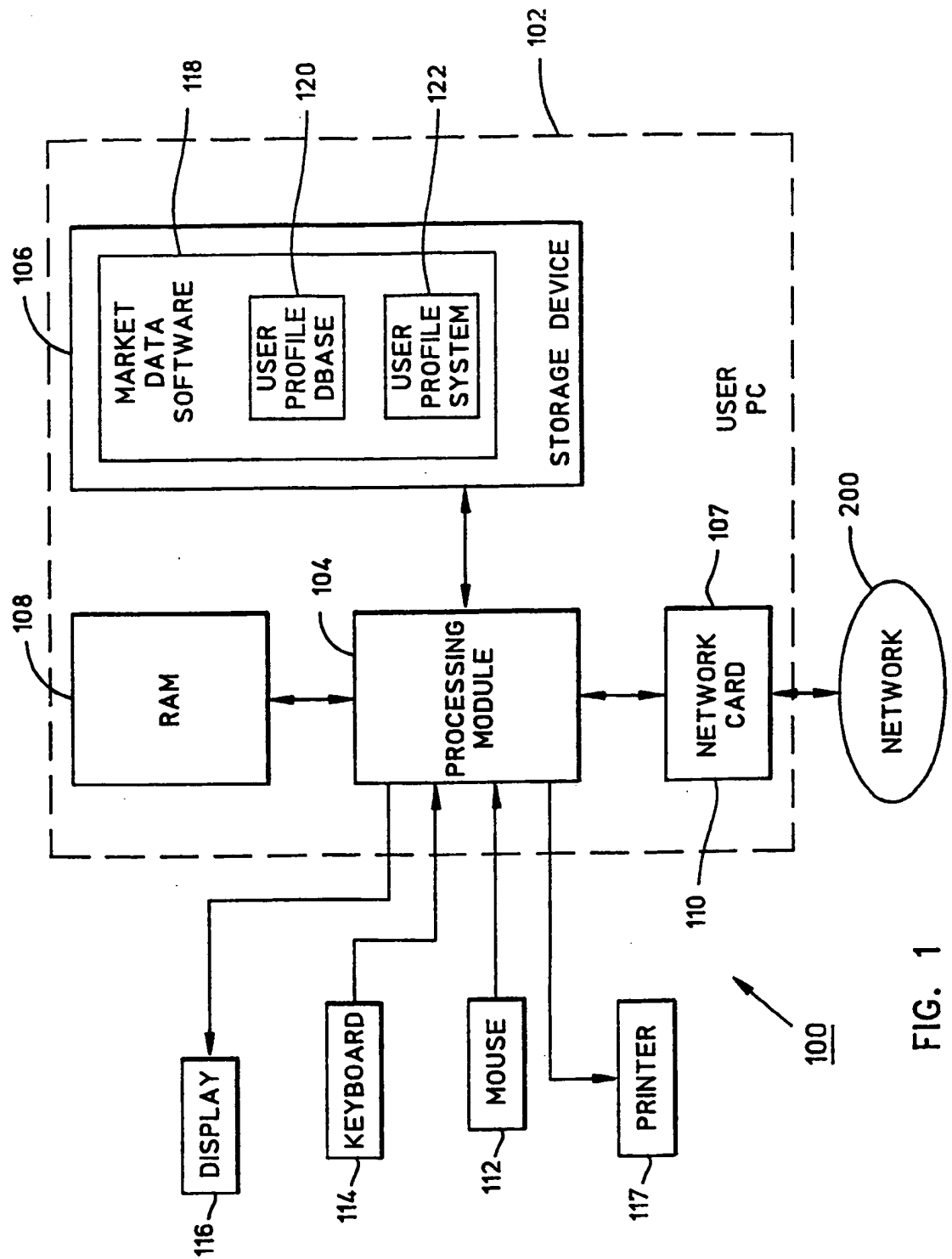


FIG. 1



2/6

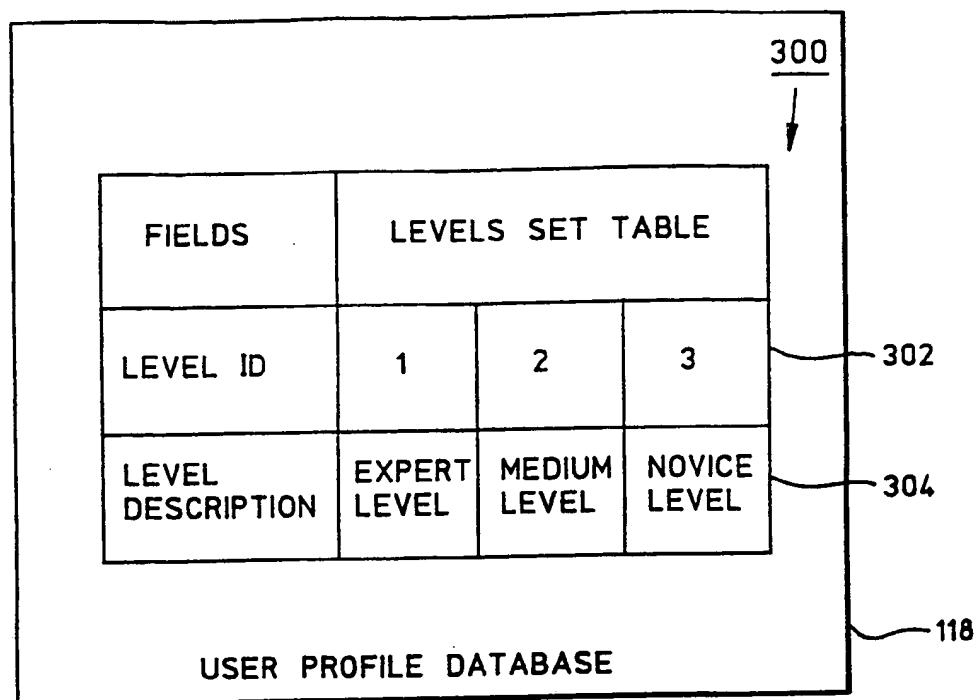


FIG. 2

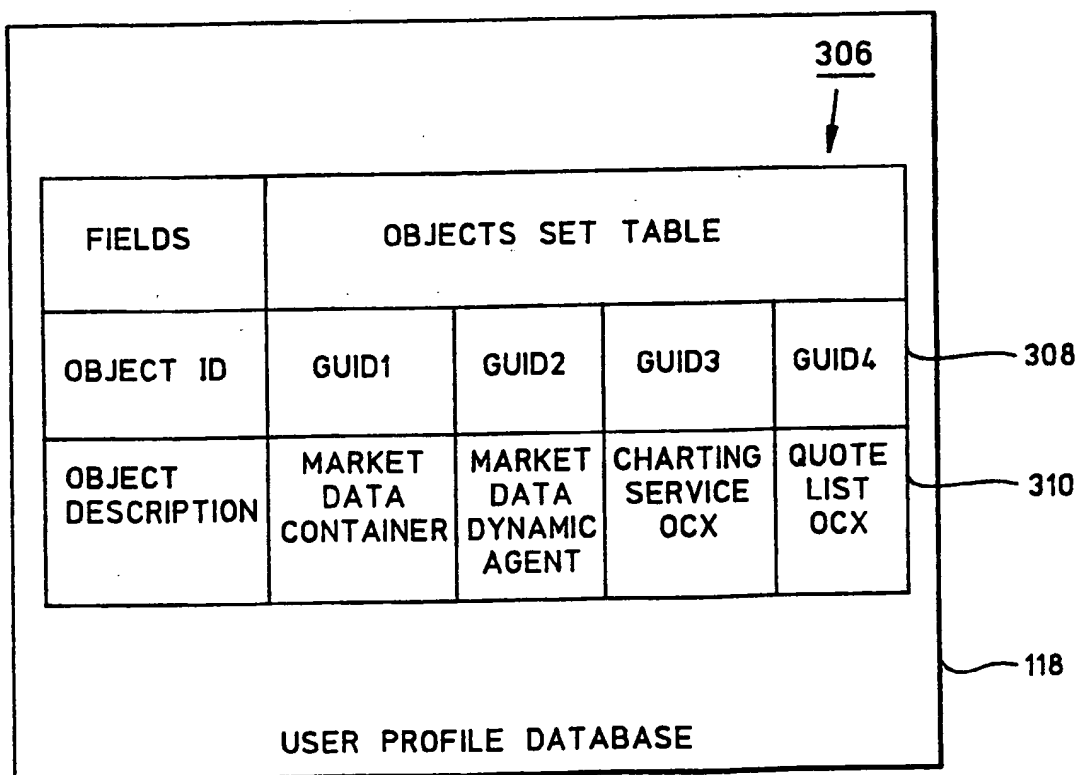


FIG. 3

LEVEL OBJECTS TABLE		
FIELDS	1	2
LEVEL ID	1	3
OBJECT ID		
GUID1	X	X
GUID2	X	X
GUID3	X	X
GUID4	X	X

USER PROFILE DATABASE

FIG. 4

OBJECT ACTIONS TABLE (FOR GUID1 MARKET DATA CONTAINER)				
OBJECT ID	GUID1	GUID1	GUID1	GUID1
ACTION ID	1	2	3	4
ACTION DESCRIPTION	DISPLAY COLOR CONTROL	NUMBER OF MODULES	NETWORK PRIORITY	DISPLAY FONT CONTROL
ACTION PURPOSE DESCRIPTION	USER CONTROL AMBIENT COLORS	#OF MODULES OPEN IN MKT DATA APPL'N	USER PRIORITY LEVEL	USER CONTROL OF FONT
ACTION TYPE ID	1	3	2	1

USER PROFILE DATABASE

FIG. 5

4/6

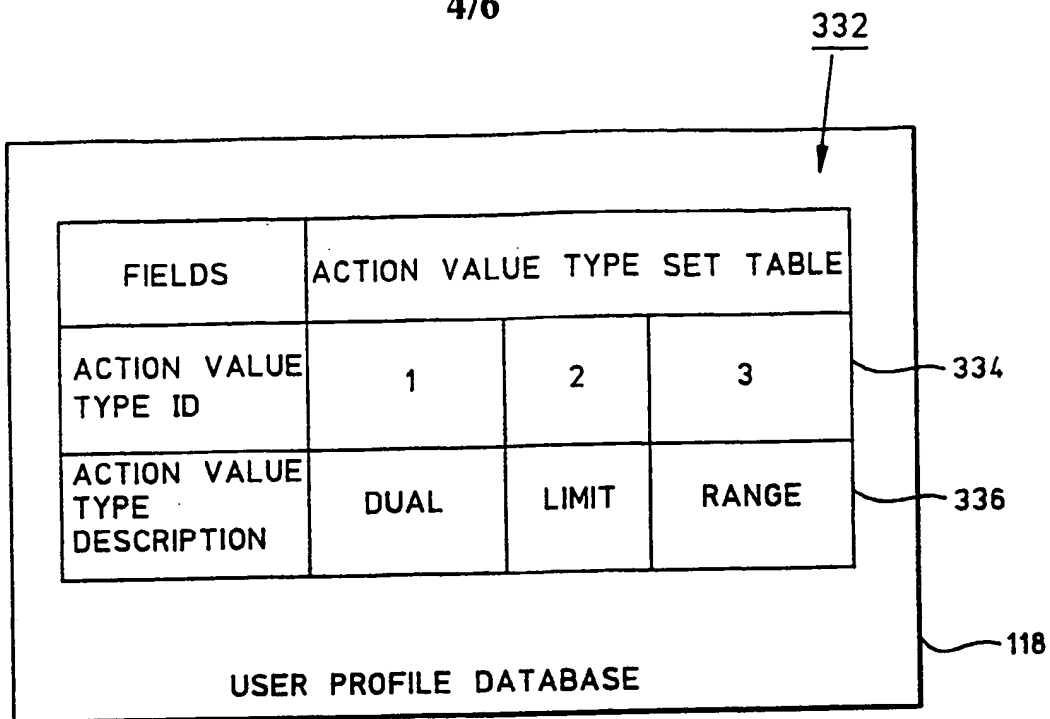


FIG. 6

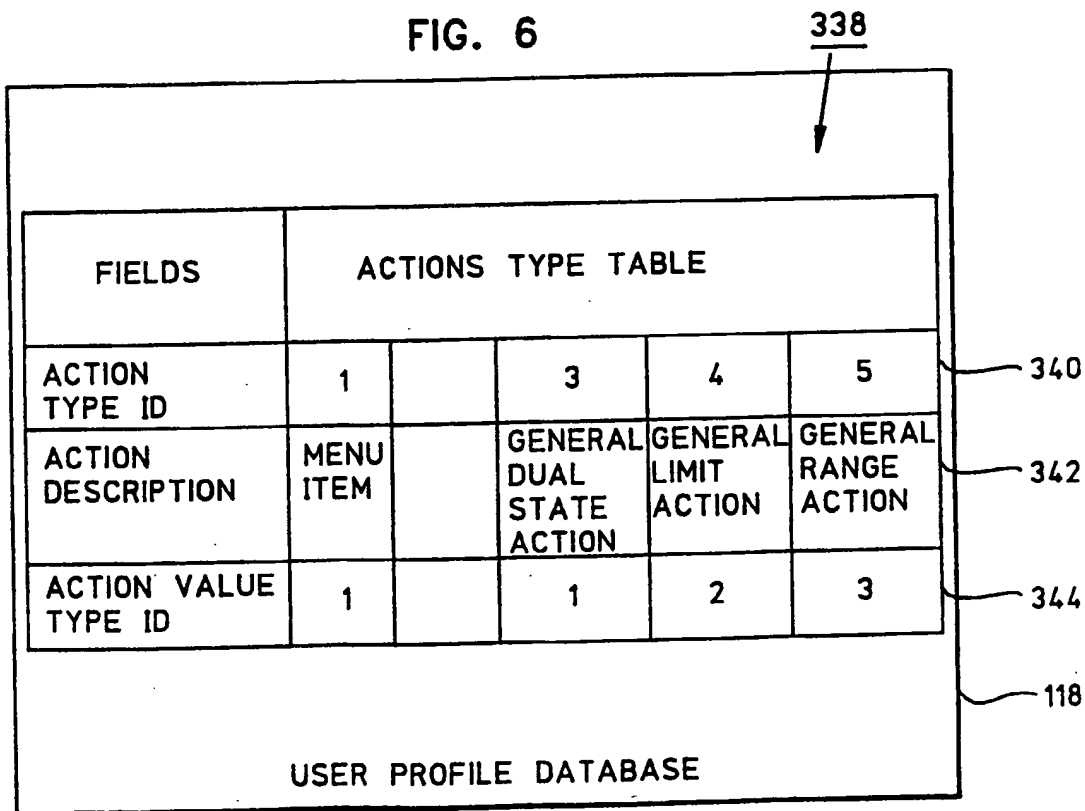


FIG. 7

5/6

346

FIELDS	MAIN PROFILE TABLE			
	LEVEL 2			
	OBJECT ID GUID1			
ACTION ID	1	2	3	4
ACTION VALUE	ENABLE	4	-	ENABLE
ACTION EXTRA VALUE	-	25	2	-

348  
350  
352  
354  
356  
118

USER PROFILE DATABASE

FIG. 8

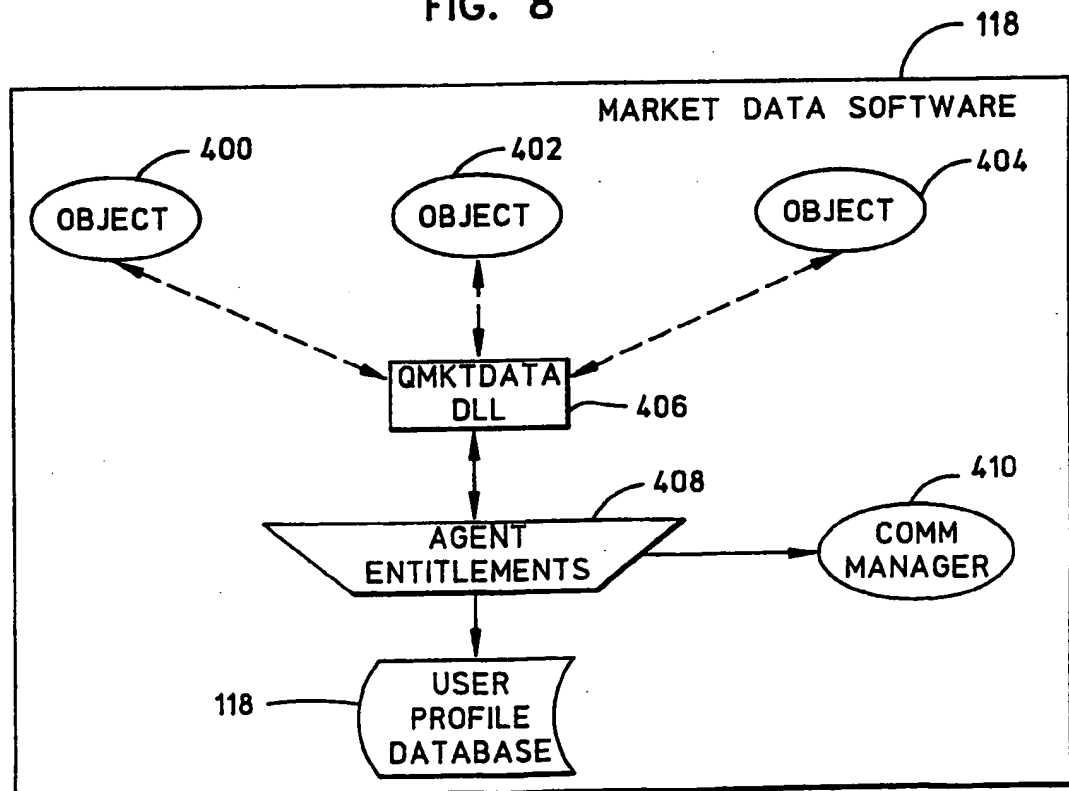


FIG. 9

6/6

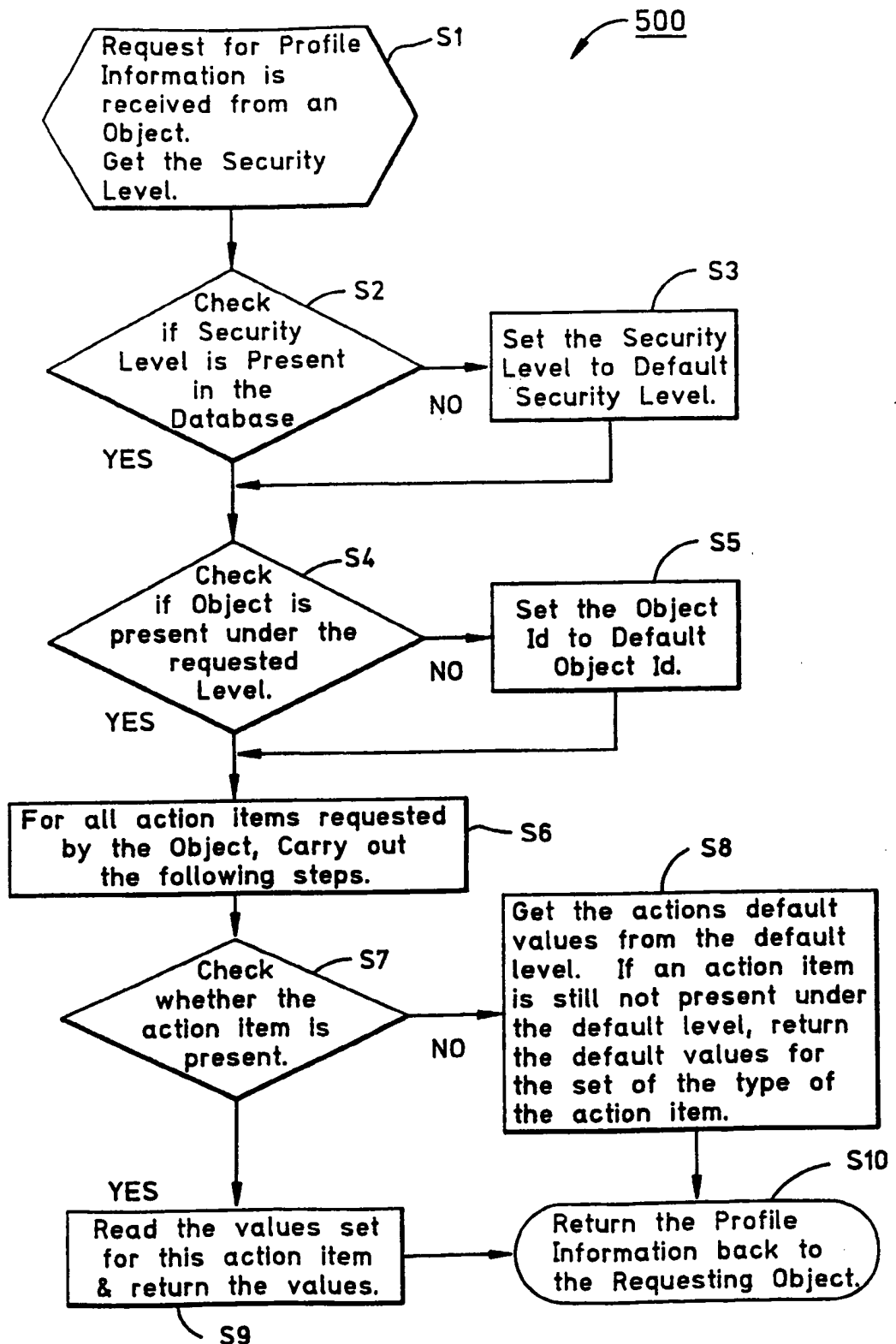


FIG. 10

# INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 99/02709

## A. CLASSIFICATION OF SUBJECT MATTER

IPC 6 G06F1/00 G06F3/033 G06F17/60

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 6 G06F H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CRIBE B E ET AL: "A COMMON DESKTOP ENVIRONMENT FOR PLATFORMS BASED ON THE UNIX OPERATING SYSTEM" HEWLETT-PACKARD JOURNAL, vol. 47, no. 2, 1 April 1996, pages 6-14, XP000591784 see paragraph 'Configuring CDE' onwards	1-14, 16-20
Y	---	15
Y	WO 94 08309 A (MARSHALL PAUL S) 14 April 1994 see abstract; figures 1,2 see page 8, last paragraph - page 9, paragraph 1 see page 12, last paragraph - page 13, paragraph 1 ---	15
	-/--	

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

### \* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search

9 June 1999

Date of mailing of the international search report

17/06/1999

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,  
Fax: (+31-70) 340-3016

Authorized officer

Powell, D

# INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 99/02709

## C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5 432 934 A (LEVIN MICHAEL ET AL) 11 July 1995 see column 1, line 12 - column 3, line 34 ---	1-20
A	US 5 677 952 A (ROGAWAY PHILLIP W ET AL) 14 October 1997 see claim 20 -----	16-20

# INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/US 99/02709

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9408309 A	14-04-1994	US 5675746 A	07-10-1997
		AU 5169493 A	26-04-1994
		CA 2146008 A	14-04-1994
		EP 0663089 A	19-07-1995
		US 5774878 A	30-06-1998
<hr/>			
US 5432934 A	11-07-1995	NONE	
<hr/>			
US 5677952 A	14-10-1997	US 5454039 A	26-09-1995
		EP 0658022 A	14-06-1995
		JP 7199808 A	04-08-1995
		SG 44363 A	19-12-1997
		US 5675652 A	07-10-1997
		US 5835597 A	10-11-1998
<hr/>			